**ĐẠI HỌC ĐÀ NẴNG**

**ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN**

**GIẢI THUẬT & LẬP TRÌNH**

**Đề tài 604 :**

**Mô phỏng cách sắp xếp hạt sỏi**

**GIẢNG VIÊN HƯỚNG DẪN: NGUYỄN THỊ MINH HỶ**

**SINH VIÊN THỰC HIỆN: NGUYỄN TRƯỜNG DUY**

**Lớp SH: 20TCLC\_KHDL**

**Đà Nẵng 6-2021**

MỤC LỤC

[MỤC LỤC 2](#_Toc493750863)

[LỜI MỞ ĐẦU 3](#_Toc493750864)

[1.GIỚI](#_Toc493750866) THIỆU ĐỀ TÀI…………………………………………………………..4

[1.1. Tên đề tài](#_Toc493750867)

[1.2. Lý do chọn đề tài](#_Toc493750868)

[1.3. Mục đích của đề tài](#_Toc493750869)

[2. CƠ SỞ LÝ THUYẾT 5](#_Toc493750870)

[2.1. Ý tưởng 5](#_Toc493750871)

[2.2. Cơ sở lý thuyết 5](#_Toc493750872)

2.2.1 Lý thuyết thuật toán Bubble sort……………………………………….

2.2.2 Cách sắp xếp được áp dụng…………………………………………….

2.2.3 Ưu điểm và nhược điểm của thuật toán bubble sort……………………

[3. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN 5](#_Toc493750873)

[3.1. Phát biểu bài toán 5](#_Toc493750874)

[3.2. Cấu trúc dữ liệu 5](#_Toc493750875)

[3.3. Thuật toán 5](#_Toc493750876)

[4. CHƯƠNG TRÌNH VÀ KẾT QUẢ 5](#_Toc493750877)

[4.1. Tổ chức chương trình 5](#_Toc493750878)

[4.2. Ngôn ngữ cài đặt 5](#_Toc493750879)

[4.3. Kết quả 5](#_Toc493750880)

[4.3.1. Giao diện chính của chương trình 5](#_Toc493750881)

[4.3.3. Nhận xét 5](#_Toc493750883)

[5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 5](#_Toc493750884)

[5.1. Kết luận 5](#_Toc493750885)

[5.2. Hướng phát triển 6](#_Toc493750886)

LỜI MỞ ĐẦU

Cùng với sự phát triển của ngành công nghệ thông tin, việc đưa máy tính vào các trường học đã tạo ra một bước ngoặc lớn trong việc dạy học. Sử dụng mô phỏng trên máy tính là phương pháp sư phạm tích cực phát huy cao độ tính độc lập, khả năng làm việc trí tuệ của sinh viên, tạo ra một nhịp độ phong cách trạng thái tâm lí mới thay đổi phương pháp và hình thức dạy và học. Chính vì vậy phương pháp mô phỏng trong giảng dạy các môn học nói chung và các môn học thuộc chuyên ngành công nghệ thông tin nói riêng được sử dụng rộng rãi ở các trường đại học trong và ngoài nước, dành cho mọi đối tượng đào tạo.

Mô phỏng thuật toán minh họa trực quan từng bước giúp người học có thể nhận thức được quá trình diễn biến và thay đổi của dữ liệu dẫn đến kết quả cuối cùng. Là cơ sở hỗ trợ đắc lực cho việc nắm và hiểu bản chất của thuật toán.

Mô phỏng thuật toán là quá trình mô tả cấu trúc dữ liệu, thao tác của một chương trình bằng đồ họa và để giúp mọi người hiểu rõ cũng như biết nhiều hơn về mô phỏng thuật toán em đã quyết định chọn đề tài “ Mô phỏng cách sắp xếp hạt sỏi”

Chương trình giải bài toán này được xây dựng dựa trên 1 thuật toán duy nhất đó là thuật toán Bubble sort (sắp xếp nổi bọt).

Em đã nghiên cứu đề tài thông qua các phương pháp:

* Phương pháp thu thập tài liệu: thu thập tài liệu từ những bài báo khoa học, các trang web tin học và một số ebook về đề tài nghiên cứu, giáo trình và các tài liệu học tập khác.
* Phương pháp phân tích và tổng hợp tài liệu: từ những tài liệu đã thu thập, tiến hành tìm hiểu, phân tích và tồng hợp nội dung liên quan đến đề tài.
* Phương pháp chuyên gia: trong quá trình nghiên cứu có sự góp ý, điều chinh từ giáo viên hướng dẫn.
* Phương pháp phân tích và tổng hợp kinh nghiệm: sau quá trình tìm hiểu và đúc kết kinh nghiệm, tiến hành tổng hợp và hoàn thiện đề tài.
* Phương pháp thực nghiệm: sau khi cài đặt xong chương trình cần xây dựng một số bộ test tiêu biểu và thực hiện kiểm tra tính đúng đắn dựa trên kết quả thu được từ chương trình.

Do kiến thức còn hạn hẹp nên chương trình của em chắc chắn sẽ còn nhiều sơ sót và hạn chế, chúng em rất mong nhận được những nhận xét góp ý chỉ bảo của thầy cô và các bạn để chương trình của chúng em có thể hoàn thiện hơn.

Em xin chân thành cảm ơn cô Nguyễn Thị Minh Hỷ đã hướng dẫn chúng em hoàn thành đề tài này.

1. **GIỚI THIỆU ĐỀ TÀI**
   1. **Tên đề tài.**

Đề tài mang tên “ Sắp xếp các hạt sỏi theo mô phỏng trực quan”.

* 1. **Lý do chọn đề tài.**

Để mang đến cho mọi người về lý thuyết cũng như là cái nhìn tổng quan về mô phỏng trực quan và thuật toán bubble sort.

* 1. **Mục đích của đề tài**

Bài toán đặt ra vấn đề khi ta có một hàng sỏi với ba màu đỏ, vàng và trắng thì chúng ta phải làm gì để sắp xếp theo một thứ tự mà chúng ta muốn dựa trên mô phỏng trực quan.

Với các yêu cầu sau:

Đầu vào: cho 1 hàng sỏi có thứ tự bất kì

**Đầu ra: sắp xếp hàng sỏi đó với thứ tự bất kì với màu giống nhau thì nằm gần nhau.**

# 2. CƠ SỞ LÝ THUYẾT

## 2.1 Ý tưởng

Quy ước các hạt sỏi là các số nguyên đại diện cho thứ tự sắp xếp mong muốn của người dùng. Giả sử người dùng chọn thứ tự sắp xếp là “vàng – trắng – đỏ” thì hạt màu vàng sẽ có giá trị 1, tương tự trắng là 2 và đỏ là 3. Việc sắp xếp các hạt sỏi trong hàng theo đúng thứ tự tương tự như việc sắp xếp mảng 1 chiều trên theo chiều từ nhỏ tới lớn.

## 2.2 Cơ sở lý thuyết

**2.2.1. Lý thuyết thuật toán Bubble sort.**

Sắp xếp nổi bọt (*bubble sort*) là một thuật toán sắp xếp đơn giản, với thao tác cơ bản là so sánh hai phần tử kề nhau, nếu chúng chưa đứng đúng thứ tự thì đổi chỗ (*swap*). Có thể tiến hành từ trên xuống (bên trái sang) hoặc từ dưới lên (bên phải sang). Sắp xếp nổi bọt còn có tên là *sắp xếp bằng so sánh trực tiếp*. Nó sử dụng phép so sánh các phần tử nên **là một giải thuật sắp xếp kiểu so sánh**.

**2.2.2. Cách sắp xếp được áp dụng.**

**Sắp xếp từ trên xuống.**

Giả sử dãy cần sắp xếp có *n* phần tử. Khi tiến hành từ trên xuống, ta so sánh hai phần tử đầu, nếu phần tử đứng trước lớn hơn phần tử đứng sau thì đổi chỗ chúng cho nhau. Tiếp tục làm như vậy với cặp phần tử thứ hai và thứ ba và tiếp tục cho đến cuối tập hợp dữ liệu, nghĩa là so sánh (và đổi chỗ nếu cần) phần tử thứ *n*-1 với phần tử thứ *n*. Sau bước này phần tử cuối cùng chính là phần tử lớn nhất của dãy.

Sau đó, quay lại so sánh (và đổi chố nếu cần) hai phần tử đầu cho đến khi gặp phần tử thứ *n*-2....

Ghi chú: Nếu trong một lần duyệt, không phải đổi chỗ bất cứ cặp phần tử nào thì danh sách đã được sắp xếp xong.

**2.2.3. Ưu điểm và nhược điểm của thuật toán Bubble sort.**

**Ưu điểm:** ưu điểm lớn nhất của Bubble sort là sự đơn giản, dễ cài đặt, sử dụng.

**Nhược điểm**: thuật toán này chạy khá chậm với tập dữ liệu có kích thước lớn.

# 3. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN

## 3.1 Phát biểu bài toán

Ta có một hàng với nhiều viên sỏi khác màu( với 3 màu: đỏ vàng trắng). Nhiệm vụ của chúng ta là sắp xếp nó theo một thứ tự bất kì mà chúng ta muốn với màu nào giống nhau thì ở gần nhau với các thứ tự như sau (đỏ-vàng-trắng, đỏ-trắng-vàng, vàng-đỏ-trắng, vàng-trắng-đỏ, trắng-vàng-đỏ, trắng-đỏ-vàng)

Yêu cầu:

Dữ liệu vào:

Một mảng 1 chiều với các giá trị 1-2-3 tương ứng với các màu sắc đỏ-vàng-trắng.

Dữ liệu ra:

Vẫn là mảng đó với giá trị thay đổi tùy vào cách sắp xếp mà bạn muốn chọn trên nền mô phỏng trực quan.

**3.2 Cấu trúc dữ liệu.**

Ta sử dụng mảng một chiều hangsoi[] để trình bày thuật toán.

## 3.3 Thuật toán sử dụng trong bài.

1. Thuật toán Bubble sort.

Giả sử có 1 mảng hangsoi[i] được khai báo toàn cục, thuật toán Bubble sort được cài đặt như sau:

for (int i = len - 1; i >= 0; i--) {

for (int j = 0; j < i; j++) {

if (hangsoi[j] > hangsoi[j + 1])

{

int temp = hangsoi[j];

hangsoi[j] = hangsoi[j + 1];

hangsoi[j + 1] = temp;

}

}

}

Trong đó:

* Len: số hạt sỏi
* Hangsoi: mảng 1 chiều đại diện cho hàng sỏi.

# CHƯƠNG TRÌNH VÀ KẾT QUẢ

## Tổ chức chương trình

Mình có 2 files .h và .cpp.

File .h để khai báo các hàm

void ve\_3\_nut\_bam();

void di\_chuyen\_2\_hat\_dau(int a[], int left, int right);

void ve\_hang\_dau\_dang\_di\_chuyen(int a[], int len, int left, int right, int after);

void reset\_game(); // reset cac variables cua game

void add\_stage(); // hien thi man hinh cho phep them hat dau

void order\_stage(); // hien thi man hinh cho phep chinh thu tu

void VeHangDau(int a[], int len);

void VeHangDauKemTheoCap(int a[], int len, int index1, int index2);

void di\_chuyen\_2\_hat\_soi(int a[], int left, int right);

void VeHangSoi(int a[], int len);

void chon\_mau(int a[], int i);

File .cpp để thực hiện các hàm đã khai báo ở trên.

#include <graphics.h>

#include <stdio.h>

#include "VeHinh.h"

#include "windows.h"

#include "mmsystem.h"

int thu\_tu\_mau\_sac[] = {

1,

2,

3

};

int so\_hat\_soi = 0;

int hangsoi[100];

const int STATE\_BEGIN = 0;

const int STATE\_ADD = 1;

const int STATE\_ORDER = 2;

int state = STATE\_ADD;

int mat\_nuoc = -1; // mat nuoc dang o sau phan tu nao?

int drawing\_color = BLACK;

int default\_radius = 30; // ban kinh cua moi hat dau trong hangdau

int default\_duong\_kinh = default\_radius \* 2;

int start\_point\_x = 100; // toa do x cuar hat dau dau tien trong hangdau

int start\_point\_y = 100; // toa do y cua hat dau dau tien trong hangdau

int default\_margin = 10; // khoang cach cac hat dau trong hangdau

char \*status = "Hay them cac vien soi";

char \*button\_text = "NEXT";

bool dachondo = false;

bool dachonvang = false;

bool dachontrang = false;

int tong\_da\_chon = 0; // tong so cac mau da chon trong luc chon thu tu

int thu\_tu\_sap\_xep[3]; // 3 1 2 => trang->do->vang

int vitri = 0; // trong luc chon hat dau, minh phai tang vi tri de biet hat dau cuoi cung o vi tri nao cua mang

void draw\_beautiful\_start\_button() {

char \*chu\_muon\_viet = button\_text;

int do\_dai\_cua\_chu = textwidth(chu\_muon\_viet);

int do\_cao\_cua\_chu = textheight(chu\_muon\_viet);

outtextxy(600-(do\_dai\_cua\_chu/2), 100 - (do\_cao\_cua\_chu/2), chu\_muon\_viet);

rectangle(500, 50, 700, 150);

}

void draw\_status() // ham xuat chu

{

outtext(status);

}

// check whether (x,y) is inside rectangle or not?

bool is\_inside\_rectangle(int x, int y, int left, int top, int right, int bottom) {

return x >= left && x <= right && y <= top && y >= bottom;

}

bool is\_inside\_start\_button(int x, int y) {

printf("click x: %d\n", x);

printf("click y: %d\n", y);

return is\_inside\_rectangle(x, y, 500, 100, 600, 200);

}

void ve\_nut\_start() {

outtextxy(500, 100, "START");

rectangle(500, 100, 600, 200);

}

void VeHangDauKemTheoCap(int a[], int len, int index1, int index2) {

int radius = default\_radius;

int duongkinh = default\_duong\_kinh;

int x = start\_point\_x;

int y = start\_point\_y + 200;

int margin = default\_margin;

for (int i = 0; i < len; i++) {

if (i > 0) {

x = x + duongkinh + margin;

}

circle(x, y, radius);

if (i == index1 || i == index2) {

line(x, y + radius, x, y + duongkinh);

}

if (mat\_nuoc != -1 && i == mat\_nuoc) {

line(x + radius + margin/2, y - radius, x + radius + margin/2, y + radius);

}

chon\_mau(a, i);

floodfill(x, y, drawing\_color);

}

}

// ve hang soi, khong ve 2 hat soi dang trong qua trinh di chuyen

void ve\_hang\_soi\_dang\_di\_chuyen(int a[], int len, int left, int right, int after) {

int radius = 30;

int duongkinh = radius \* 2;

int x = start\_point\_x;

int y = start\_point\_y + 200;

int margin = default\_margin;

for (int i = 0; i < len; i++) {

if (i > 0) {

x = x + duongkinh + margin;

}

if (i == left ||i == right) {

continue;

}

if (mat\_nuoc != -1 && i == mat\_nuoc) {

line(x + radius + margin/2, y - radius, x + radius + margin/2, y + radius);

}

circle(x, y, radius);

chon\_mau(a, i);

floodfill(x, y, drawing\_color);

}

}

// ve hang soi sau khi da bien doi mang a

void VeHangSoi(int a[], int len) {

int radius = default\_radius;

int duongkinh = default\_duong\_kinh;

int x = start\_point\_x;

int y = start\_point\_y + 200;

int margin = default\_margin;

for (int i = 0; i < len; i++) {

if (i > 0) {

x = x + duongkinh + margin;

}

if (mat\_nuoc != -1 && i == mat\_nuoc) {

line(x + radius + margin/2, y - radius, x + radius + margin/2, y + radius);

}

circle(x, y, radius);

chon\_mau(a, i);

floodfill(x, y, drawing\_color);

}

}

void thuat\_toan\_theo\_thu\_tu(int thu\_tu[]) {

status = "Dang sap xep";

int len = so\_hat\_soi;

thu\_tu\_mau\_sac[0] = thu\_tu[0];

thu\_tu\_mau\_sac[1] = thu\_tu[1];

thu\_tu\_mau\_sac[2] = thu\_tu[2];

// thay doi cac gia tri theo dung thu tu cua no

for (int i = 0; i < len; i++) {

// 1 do

// 2 vang

// 3 trang

// 2 vang

//

// thu tu: vang, do, trang [2, 1, 3]

//

// [1, 2, 3, 2] => [2, 1, 3, 1] -- sap xep=> [1, 1, 2, 3] <=> [vang, vang, do, trang]

if (hangsoi[i] == thu\_tu[0]) {

hangsoi[i] = 1;

} else if (hangsoi[i] == thu\_tu[1]) {

hangsoi[i] = 2;

} else {

hangsoi[i] = 3;

}

}

for (int i = len - 1; i >= 0; i--) {

mat\_nuoc = i;

for (int j = 0; j < i; j++) {

cleardevice();

draw\_status();

ve\_3\_nut\_bam();

VeHangDauKemTheoCap(hangsoi, len, j, j + 1);

delay(1000);

cleardevice();

if (hangsoi[j] > hangsoi[j + 1]) {

di\_chuyen\_2\_hat\_soi(hangsoi, j, j+1);

int temp = hangsoi[j];

hangsoi[j] = hangsoi[j + 1];

hangsoi[j + 1] = temp;

}

cleardevice();

draw\_status();

ve\_3\_nut\_bam();

VeHangSoi(hangsoi, len);

delay(1000);

}

}

}

void chon\_mau(int a[], int i) {

int x = 100;

int y = 100;

int radius = 50;

int khoang\_cach = 10;

int mau\_nguyen\_thuy = thu\_tu\_mau\_sac[a[i] - 1];

if (mau\_nguyen\_thuy == 1) {

setfillstyle(SOLID\_FILL, RED);

} else if (mau\_nguyen\_thuy == 2) {

setfillstyle(SOLID\_FILL, YELLOW);

} else {

setfillstyle(SOLID\_FILL, WHITE);

}

}

bool kiem\_tra\_xy\_nam\_trong\_hinh\_tron(int x, int y, int tamX, int tamY, int radius) {

return ((x >= tamX - radius) &&

x <= tamX + radius &&

y >= tamY - radius &&

y <= tamY + radius);

}

bool kiem\_tra\_xy\_trong\_hinh\_vuong(int left,int top,int right,int bottom,int x, int y)

{

return(( x > left)

&& (x < right)

&& ( y > top)

&& (y < bottom));

}

void ve\_3\_nut\_bam() {

draw\_beautiful\_start\_button();

int radius = 50;

// ve hat dau mau do

circle(100, 100, radius);

setfillstyle(SOLID\_FILL, RED);

floodfill(100, 100, drawing\_color);

circle(210, 100, radius);

setfillstyle(SOLID\_FILL, YELLOW);

floodfill(210, 100, drawing\_color);

circle(320, 100, radius);

setfillstyle(SOLID\_FILL, WHITE);

floodfill(320, 100, drawing\_color);

}

void VuaThemVuaVeHangSoi(int a[], int len) {

int radius = 30;

int duongkinh = radius \* 2;

int x = start\_point\_x;

int y = start\_point\_y + 200;

for (int i = 0; i < len; i++) {

circle(x + i \* (duongkinh + 10), y, radius);

// 2 ham duoi to mau

if (a[i] == 1) {

setfillstyle(SOLID\_FILL, RED);

} else if (a[i] == 2) {

setfillstyle(SOLID\_FILL, YELLOW);

} else {

setfillstyle(SOLID\_FILL, WHITE);

}

floodfill(x + i \* (duongkinh + 10), y, drawing\_color);

}

}

void di\_chuyen\_2\_hat\_soi(int a[], int left, int right) {

int x = start\_point\_x;

int y = start\_point\_y + 200;

int radius = 30;

int khoang\_cach = 10;

int x\_left = x + left \* (radius \* 2 + khoang\_cach);

int y\_left = y;

int x\_right = x + right \* (radius \* 2 + khoang\_cach);

int y\_right = y;

int do\_dai = x\_right - x\_left;

int do\_dai\_moi\_buoc = 10;

int so\_buoc = do\_dai / do\_dai\_moi\_buoc;

int nhich\_len = radius + 1;

y\_left = y\_left + nhich\_len;

y\_right = y\_right - nhich\_len;

cleardevice();

draw\_status();

ve\_3\_nut\_bam();

ve\_hang\_soi\_dang\_di\_chuyen(a, so\_hat\_soi, left, right, so\_hat\_soi - 1);

// hinh left nhich len, right di xuong

circle(x\_left, y\_left, radius);

chon\_mau(a, left);

floodfill(x\_left, y\_left, drawing\_color);

circle(x\_right, y\_right, radius);

chon\_mau(a, right);

floodfill(x\_right, y\_right, drawing\_color);

delay(300);

// left va right di ngang qua lai

for (int i = 1; i <= so\_buoc; i++) {

cleardevice();

draw\_status();

ve\_3\_nut\_bam();

ve\_hang\_soi\_dang\_di\_chuyen(a, so\_hat\_soi, left, right, mat\_nuoc);

x\_left = x\_left + do\_dai\_moi\_buoc;

circle(x\_left, y\_left, radius);

chon\_mau(a, left);

floodfill(x\_left, y\_left, drawing\_color);

x\_right = x\_right - do\_dai\_moi\_buoc;

circle(x\_right, y\_right, radius);

chon\_mau(a, right);

floodfill(x\_right, y\_right, drawing\_color);

delay(300);

}

cleardevice();

draw\_status();

ve\_3\_nut\_bam();

ve\_hang\_soi\_dang\_di\_chuyen(a, so\_hat\_soi, left, right, mat\_nuoc);

// hinh left di xuong, right di len

y\_left = y\_left - nhich\_len;

circle(x\_left, y\_left, radius);

chon\_mau(a, left);

floodfill(x\_left, y\_left, drawing\_color);

circle(x\_right, y\_right, radius);

y\_right = y\_right + nhich\_len;

chon\_mau(a, right);

floodfill(x\_right, y\_right, drawing\_color);

}

bool should\_render\_for\_add\_stage = true;

void render\_for\_add\_stage() {

if (should\_render\_for\_add\_stage) {

cleardevice();

draw\_status();

ve\_3\_nut\_bam();

VuaThemVuaVeHangSoi(hangsoi, so\_hat\_soi);

should\_render\_for\_add\_stage = false;

}

delay(10);

}

void add\_stage() {

button\_text = "TIEP THEO";

int x, y;

while (TRUE) {

render\_for\_add\_stage();

if (ismouseclick(WM\_LBUTTONDOWN)) {

getmouseclick(WM\_LBUTTONDOWN, x, y);

if (kiem\_tra\_xy\_nam\_trong\_hinh\_tron(x, y, 100, 100, 50)) {

PlaySound(TEXT("cach.wav"),NULL,SND\_SYNC);

status = "THEM HAT SOI: ban da click vien soi DO";

hangsoi[vitri] = 1;

vitri++;

so\_hat\_soi++;

should\_render\_for\_add\_stage = true;

} else if (kiem\_tra\_xy\_nam\_trong\_hinh\_tron(x, y, 210, 100, 50)) {

PlaySound(TEXT("cach.wav"),NULL,SND\_SYNC);

status = "THEM HAT SOI: ban da click vien soi VANG";

hangsoi[vitri] = 2;

so\_hat\_soi++;

vitri++;

should\_render\_for\_add\_stage = true;

} else if (kiem\_tra\_xy\_nam\_trong\_hinh\_tron(x, y, 320, 100, 50)) {

PlaySound(TEXT("cach.wav"),NULL,SND\_SYNC);

status = "THEM HAT SOI: ban da click vien soi TRANG";

hangsoi[vitri] = 3;

so\_hat\_soi++;

vitri++;

should\_render\_for\_add\_stage = true;

} else { // neu click vao nut START

if (kiem\_tra\_xy\_trong\_hinh\_vuong(500,50,700,150,x,y) ){

// kiem tra hang dau da chon duoc phan tu nao chua?

if (so\_hat\_soi == 0)

{

status = "Vui long chon it nhat 1 hat soi de bat dau";

should\_render\_for\_add\_stage = true;

continue;

}

status = "Da click vao nut next. Ban muon xep theo thu tu nao?";

should\_render\_for\_add\_stage = true;

break;

}

}

}

}

}

bool should\_render\_for\_order\_stage = true;

void render\_for\_order\_stage() {

if (should\_render\_for\_order\_stage) {

cleardevice();

draw\_status();

ve\_3\_nut\_bam();

// ve hang dau

VuaThemVuaVeHangSoi(hangsoi, so\_hat\_soi);

should\_render\_for\_order\_stage = false;

}

delay(10);

}

void reset\_game() {

vitri = 0;

so\_hat\_soi = 0;

status = "Them hat soi bat ki ma ban muon";

tong\_da\_chon = 0;

dachondo = false;

dachonvang = false;

dachontrang = false;

}

void draw\_thu\_tu() {

char tt[1000];

char x[] = "THU TU:";

strncat(tt, x, 7);

for (int i = 0; i < tong\_da\_chon; i++) {

switch(thu\_tu\_sap\_xep[i])

{

case 1:

{

char content[] = "->DO";

strncat(tt, content, strlen(content));

break;

}

case 2:

{

char content[] = "->VANG";

strncat(tt, content, strlen(content));

break;

}

case 3:

{

char content[] = "->TRANG";

strncat(tt, content, strlen(content));

break;

}

}

}

status = tt;

}

void order\_stage() {

should\_render\_for\_order\_stage = true;

button\_text = "BAT DAU";

int x, y;

while (TRUE) {

render\_for\_order\_stage();

if (tong\_da\_chon == 3) {

// outtext("nhan nut start");

// wait for click start button. then exit (break)

while(TRUE) {

if (ismouseclick(WM\_LBUTTONDOWN)) {

getmouseclick(WM\_LBUTTONDOWN, x, y);

if (kiem\_tra\_xy\_trong\_hinh\_vuong(500, 50,600,150,x,y) ){

break;

}

}

delay(10);

}

break;

}

if (ismouseclick(WM\_LBUTTONDOWN)) {

getmouseclick(WM\_LBUTTONDOWN, x, y);

if (kiem\_tra\_xy\_nam\_trong\_hinh\_tron(x, y, 100, 100, 50)) {

if (!dachondo) {

thu\_tu\_sap\_xep[tong\_da\_chon] = 1;

dachondo = true;

tong\_da\_chon++;

draw\_thu\_tu();

should\_render\_for\_order\_stage = true;

}

} else if (kiem\_tra\_xy\_nam\_trong\_hinh\_tron(x, y, 210, 100, 50)) {

if (!dachonvang) {

thu\_tu\_sap\_xep[tong\_da\_chon] = 2;

dachonvang= true;

tong\_da\_chon++;

draw\_thu\_tu();

should\_render\_for\_order\_stage = true;

}

} else if (kiem\_tra\_xy\_nam\_trong\_hinh\_tron(x, y, 320, 100, 50)) {

if (!dachontrang) {

thu\_tu\_sap\_xep[tong\_da\_chon] = 3;

dachontrang = true;

tong\_da\_chon++;

draw\_thu\_tu();

should\_render\_for\_order\_stage = true;

}

}

}

}

cleardevice();

// tien hanh sap xep

thuat\_toan\_theo\_thu\_tu(thu\_tu\_sap\_xep);

}

int last\_screen()

{ int x,y;

setusercharsize(2,1,2,1);

rectangle(200,50,400,150);

char \*chu\_muon\_viet1 = "QUAY LAI";

int do\_dai\_cua\_chu1 = textwidth(chu\_muon\_viet1);

int do\_cao\_cua\_chu1 = textheight(chu\_muon\_viet1);

outtextxy(300-(do\_dai\_cua\_chu1/2), 100 - (do\_cao\_cua\_chu1/2), chu\_muon\_viet1);

rectangle(600,50,800,150);

char \*chu\_muon\_viet2 = "KET THUC";

int do\_dai\_cua\_chu2 = textwidth(chu\_muon\_viet2);

int do\_cao\_cua\_chu2 = textheight(chu\_muon\_viet2);

outtextxy(700-(do\_dai\_cua\_chu2/2), 100 - (do\_cao\_cua\_chu2/2), chu\_muon\_viet2);

while(TRUE) {

if (ismouseclick(WM\_LBUTTONDOWN)) {

getmouseclick(WM\_LBUTTONDOWN, x, y);

if(kiem\_tra\_xy\_trong\_hinh\_vuong(600, 50, 800, 150, x, y))

{

return 1;

}

if(kiem\_tra\_xy\_trong\_hinh\_vuong(200, 50, 400, 150, x, y))

{

return 2;

}

}

delay(100);

}

}

void render\_game() {

initwindow(1000, 500, "ten window");

setusercharsize(2, 1, 2, 1);

setbkcolor(CYAN); // mau nen

drawing\_color = BLACK;

setcolor(drawing\_color); // drawing color == mau chu

while(TRUE){

switch(state) {

case STATE\_BEGIN:

{

reset\_game();

state = STATE\_ADD;

}

case STATE\_ADD:

{

add\_stage();

state = STATE\_ORDER;

}

case STATE\_ORDER:

{

order\_stage();

cleardevice();

int flag = last\_screen();

if (flag == 1) {

closegraph();

break;

}

if (flag == 2) {

state = STATE\_BEGIN;

continue;

}

}

default:

{

break;

}

}

}

}

int main() {

render\_game();

return 0;

}

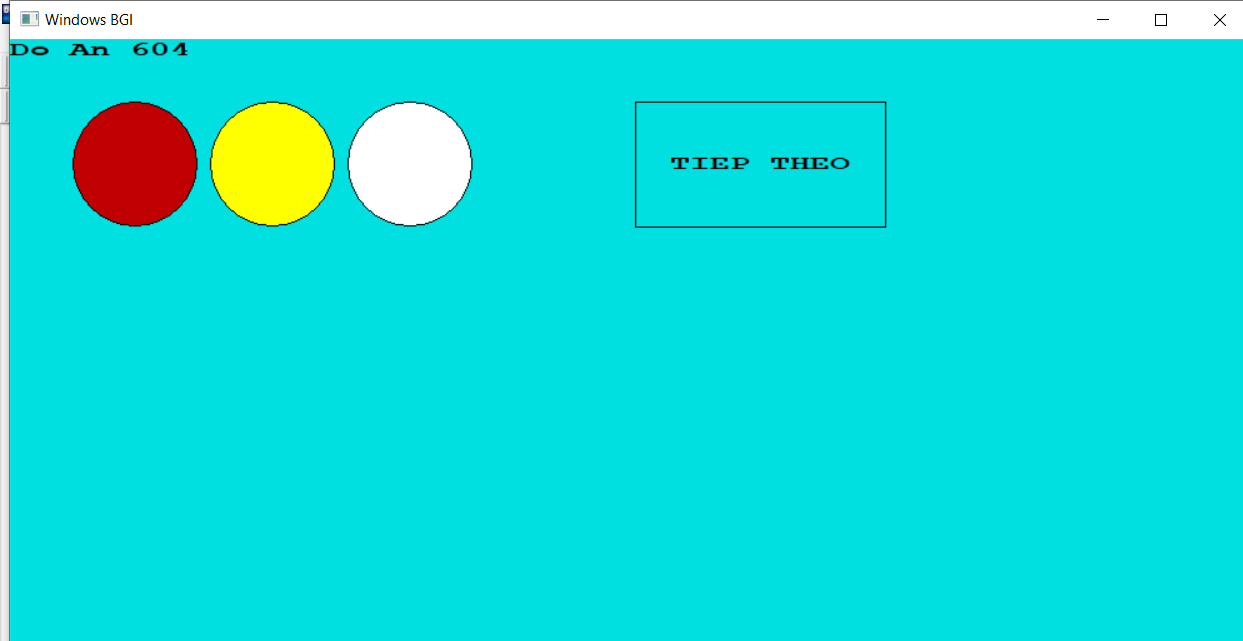
## Ngôn ngữ cài đặt

Ngôn ngữ lập trình C, sử dụng thư viện Borland C Graphics Interface (BGI).

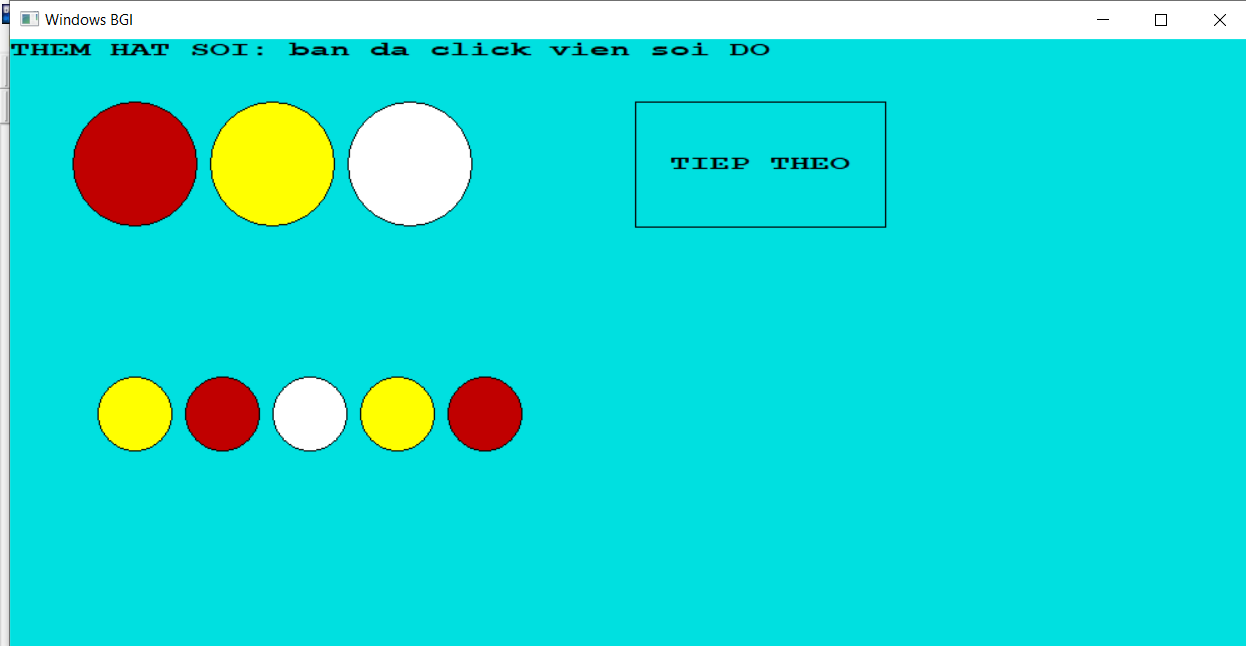
## Kết quả

### Giao diện của chương trình

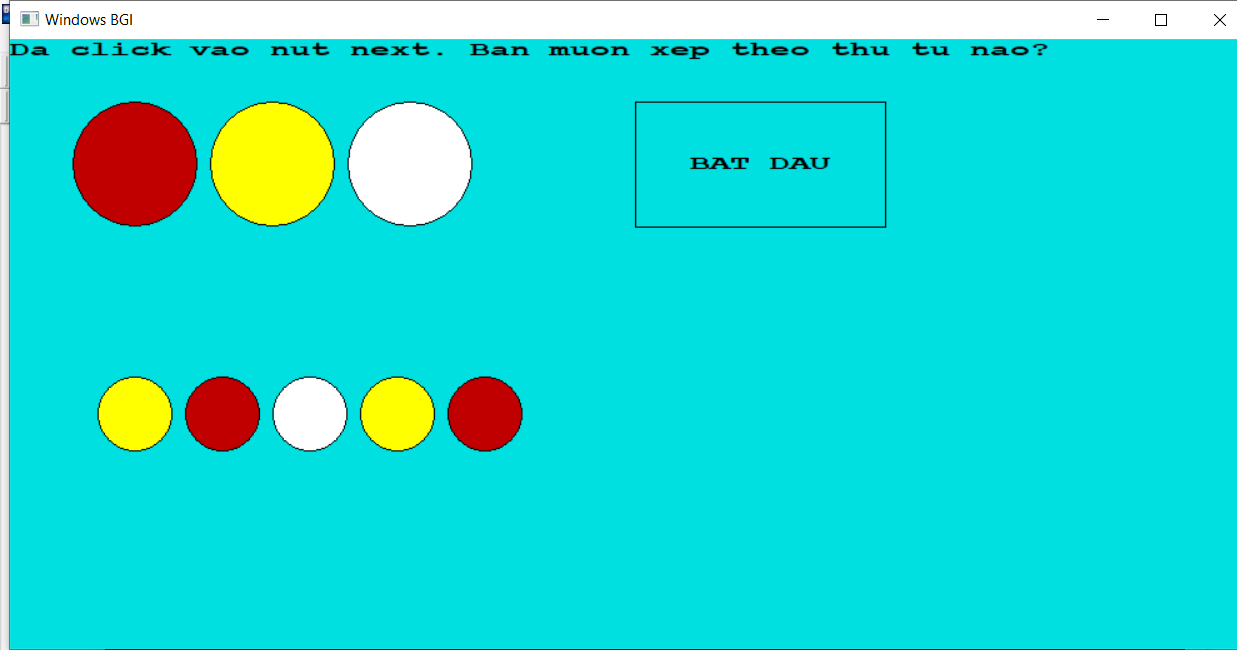
1. Đây là màn hình đầu tiên để thêm hạt sỏi vào hàng.



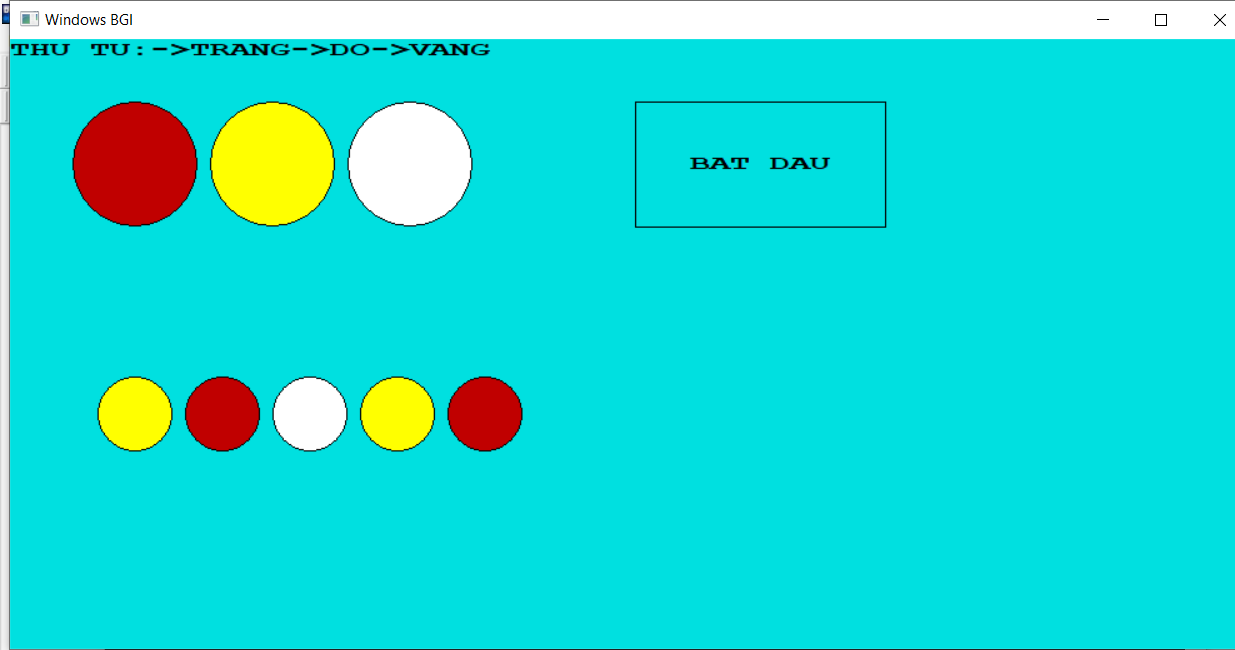
1. Đây là màn hình đã thêm vào các hạt sỏi vào hàng với thứ tự ban đầu như hình.



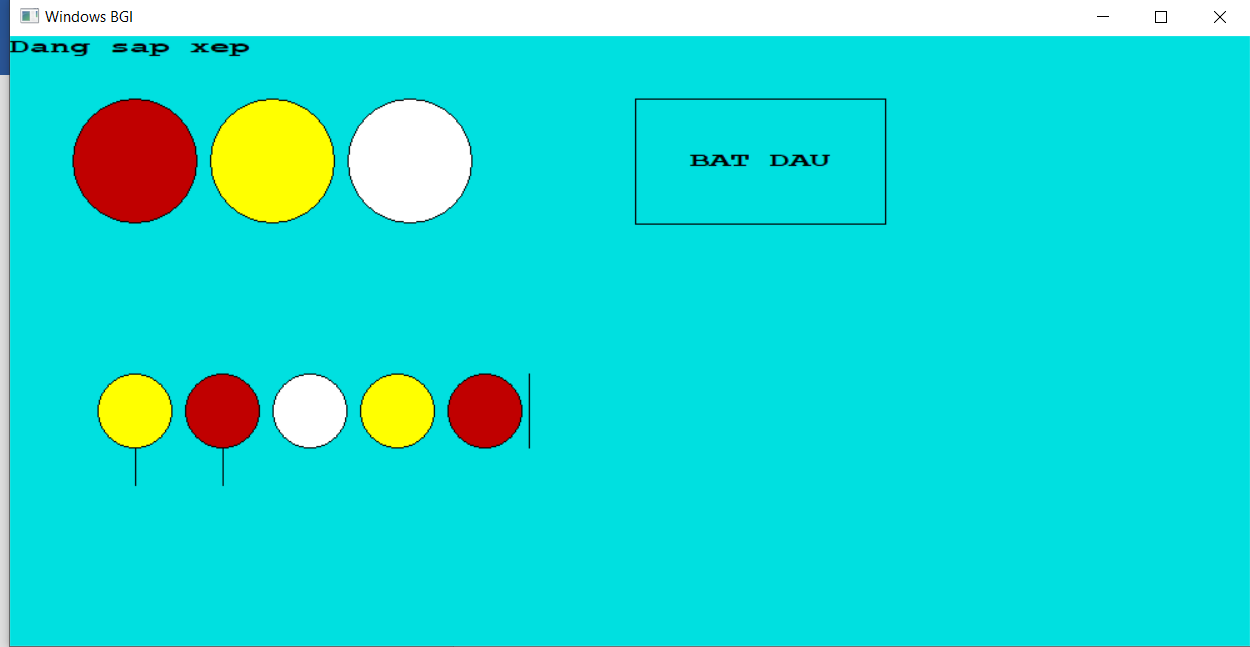
1. Sau khi bấm nút next thì được chuyển đến màn hình tiếp theo là màn hình chọn thứ tự màu sắc cần sắp xếp. Chọn đủ 3 màu bằng cách bấm chuột trái vào các hạt đậu theo thứ tự mà bạn muốn.



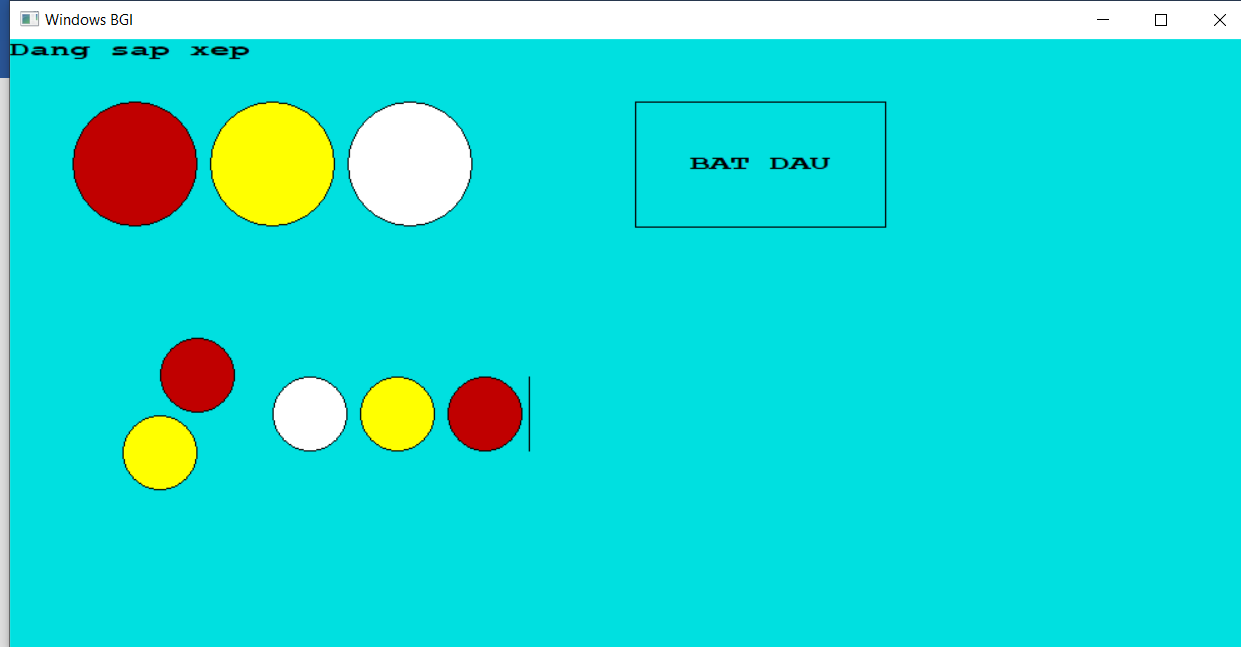
1. Theo màn hình thì ta đã chọn thứ tự màu sỏi cần sắp xếp là trắng -> đỏ -> vàng, sau đó ấn nút START để đến màn hình mô phỏng thuật toán bubble sort.



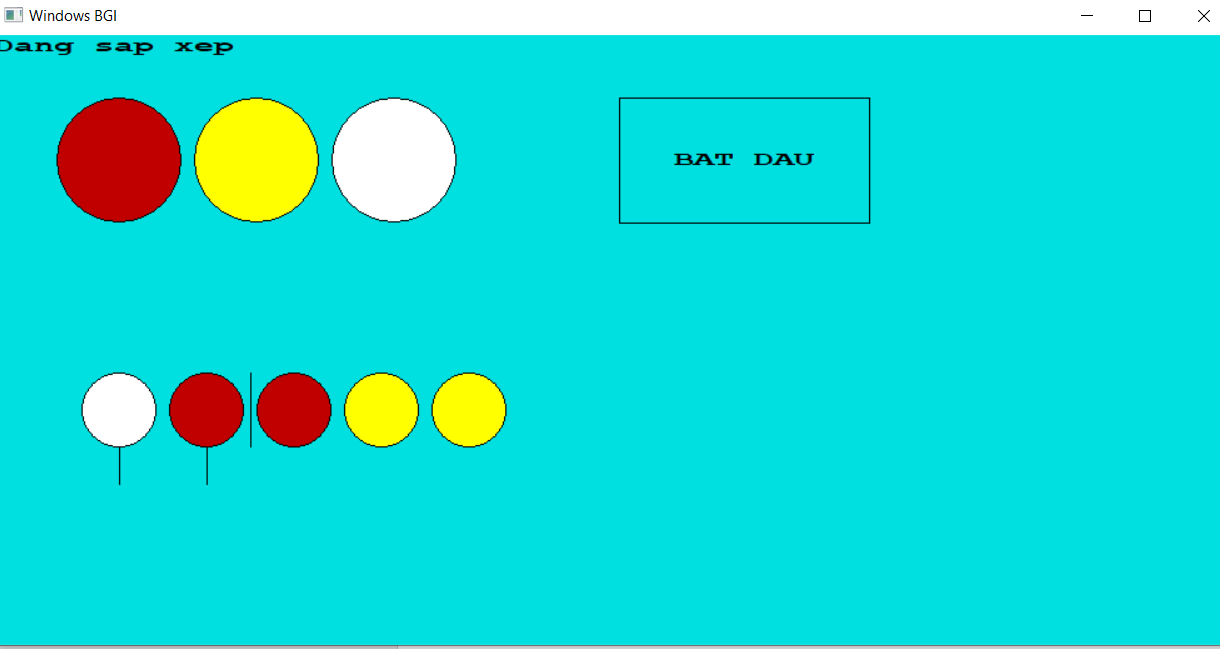
1. Đây là màn hình bắt đầu sắp xếp các hạt sỏi theo thứ tự đã chọn. Trong hình có 1 đường thẳng đứng ngắn cách các hạt sỏi để mô phỏng mặt nước trong thuật toán bubble sort, 2 đường thẳng đứng khác có đỉnh nối với 2 hạt sỏi để chỉ rằng 2 hạt sỏi đó đang được so sánh, nếu thứ tự của nó không đúng thứ tự sắp xếp thì sẽ hoán đổi.



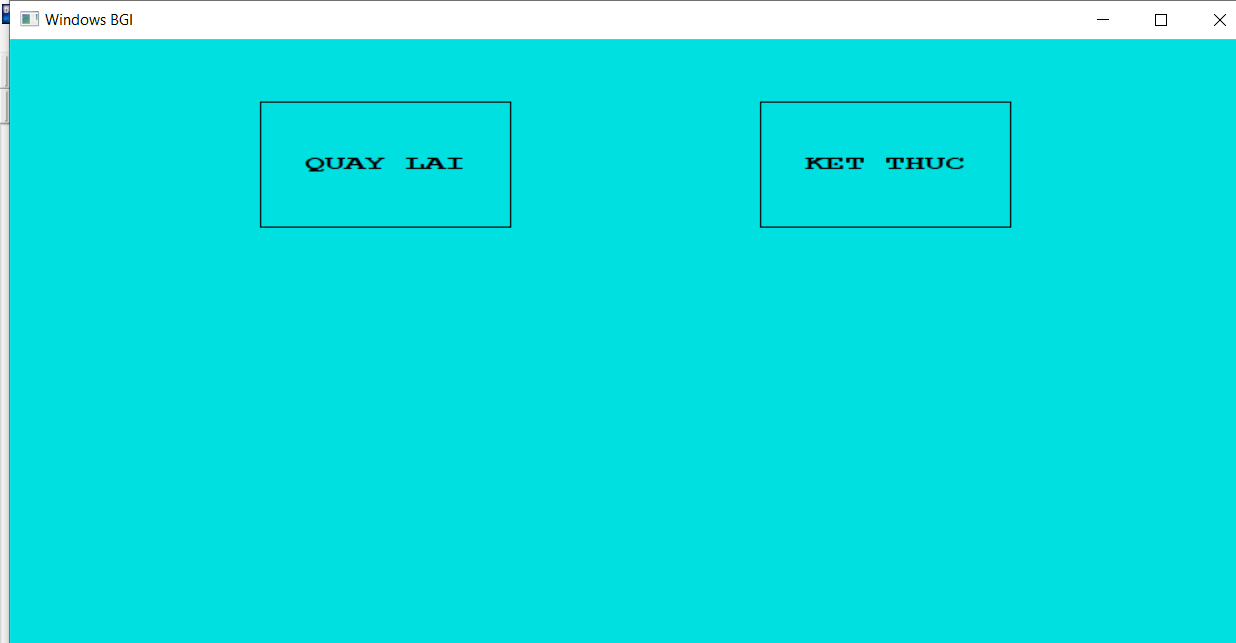
Hạt sỏi màu đỏ ở trước, và vàng ở sau không đúng thứ tự (vàng - đỏ - trắng), nên nó sẽ đổi chỗ cho nhau, trong hình mô phỏng việc đổi chỗ giữa chúng.



1. Đây là màn hình đã sắp xếp hoàn tất các hạt sỏi về đúng thứ tự mà mình muốn sắp xếp (trắng - vàng - đỏ).



1. Và đây là màn hình cuối cùng cho ta hai sự lựa chọn đó là tiếp tục hay kết thúc chương trình.



### Nhận xét

* Kết quả của bài toán chính xác .
* Giao diện mô phỏng trực quan rõ ràng và dễ hiểu.
* Độ phức tạp của bài toán thấp nhưng hiệu suất thì vẫn còn thấp.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

* Chương trình đáp ứng tất cả các nhu cầu của đề bài đưa ra.
* Giao diện của chương trình dễ nhìn, dễ hiểu và dễ thực hiện.
* Độ phức tạp thấp nhưng hiệu suất của bài toán vẫn còn thấp.
* Mở rộng thêm bài toán yêu cầu trong thực tế.

## Hướng phát triển

* Bài toán mới chỉ giải quyết được số hạt sỏi còn quá bé nếu muốn tăng tốc độ tính toán và tăng số lượng hạt sỏi lớn hơn thì phải tăng độ nhanh của thuật toán sắp xếp.
* Có thể đa dạng nhiều màu sắc hơn là ba màu trong bài.
* Có thể đa dạng cách sắp xếp hơn và đa dạng hình thù của viên sỏi.